



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/840,167	05/06/2004	Nicolai Kosche	188377/US/2	7505
66083	7590	12/13/2007	EXAMINER	
SUN MICROSYSTEMS, INC. c/o DORSEY & WHITNEY, LLP			STEELMAN, MARY J	
370 SEVENTEENTH ST.				
SUITE 4700			ART UNIT	PAPER NUMBER
DENVER, CO 80202			2191	
MAIL DATE		DELIVERY MODE		
12/13/2007		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>
	10/840,167	KOSCHE ET AL.
	<b>Examiner</b>	<b>Art Unit</b>
	MARY STEELMAN	2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) Responsive to communication(s) filed on 06 May 2004.
- 2a) This action is FINAL.                    2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) Claim(s) 1-52 is/are pending in the application.
  - 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) Claim(s) \_\_\_\_\_ is/are allowed.
- 6) Claim(s) 1-52 is/are rejected.
- 7) Claim(s) \_\_\_\_\_ is/are objected to.
- 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.
 

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
  - a) All
  - b) Some \*
  - c) None of:
    1. Certified copies of the priority documents have been received.
    2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
    3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____
3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)	5) <input type="checkbox"/> Notice of Informal Patent Application
Paper No(s)/Mail Date <u>05/06/04, 03/17/06</u> .	6) <input type="checkbox"/> Other: _____

**DETAILED ACTION**

1. Claims 1-52 are pending.

*Specification*

2. The use of the trademark JAVA has been noted in this application. It should be capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

The disclosure is objected to because it contains an embedded hyperlink and/or other form of browser-executable code. Applicant is required to delete the embedded hyperlink and/or other form of browser-executable code. See MPEP § 608.01.

3. Remove the hyperlink from the Specification at [1007].

*IDS*

4. IDS received 05/06/2004 has been considered.

5. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-15, 24, 38, 40-45 & 49-52 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 1-15 recite “a software tool”, which is software per se, a non statutory embodiment.

Claims 24 & 38 recite “machine readable media”, defined in the Specification at [1095] to include propagated signal... These are non-statutory embodiments.

Independent claim 40, recites “a computer program product”, defined in the Specification at [1095] to include propagated signal...carrier waves. These are non-statutory embodiments.

Claims 49-52 recite “machine readable media”, defined in the Specification at [1095] to include propagated signal... These are non-statutory embodiments.

***Claim Rejections - 35 USC § 103***

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Art Unit: 2191

7. Claims 1-52 are rejected under 35 U.S.C. 103(a) as being unpatentable over USPN 7,103,877 to Arnold et al., in view of USPN 6,374,367 B1 to Dean et al.

Per claim 1:

A software tool that, based at least in part on a predefined association between an instruction instance in executable code and a representation of a source-level data object language construct corresponding thereto, attributes runtime events to source-level data objects.

Arnold disclosed:

Col. 2: 19-22, "system and method for collecting information...a statistical sample of information..." Col. 5: 13-29, "sampling information may be collected...a low level mechanism exists that is available to map from a taken yield point to a method...inspecting the hardware state to determine the instruction address...mapping that address to a method...inspecting the program's runtime stack to identify the method...inspecting the return addresses stored...identify and track executing methods...for characterizing program behavior..."

More explicitly, Dean disclosed (col. 5: 31-40), sampling...collecting state information. The stored information (FIG. 3) includes an address field 320 storing the virtual and / or physical addresses associated with the transaction. Col. 7: 35-43, consideration given to cache memory

Art Unit: 2191

regions to estimate conflicts (cache misses) Col. 11: 8, get a precise understanding of the internal operation of an out of order issue processor. Col. 11: 41, statistically reconstruct detailed processor pipeline states.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Arnold, using the teachings of Dean, because both references are directed towards profiling for optimization, using identified statistical analysis, related to cache misses / coherency, receiving feedback to improve program performance.

Per claim 2:

-the runtime events include sampled runtime events that statistically represent the runtime events.

Arnold: Col. 2:22, "statistical sample of the information..."

Per claim 3:

-the sampled runtime events include one or more of cache misses, cache references, data translation buffer misses, data translation buffer references, traps, and an event counter condition.

Arnold: Col. 3: 55-67, "...interrupt (trap) is arranged to occur at periodic time intervals. An interrupt handler is coded to catch the timer interrupt..." Col. 4: 19, "counter based approach" (counter condition)

Per claim 4:

-the event counter condition includes counter underflow or counter overflow.

Arnold disclosed a (Col. 4: 17) decrementing counter and (col. 4: 43-44) a count register and a decrement and conditional branch on count instruction. Arnold disclosed (col. 8: 31-36), initializing a counter to a predetermined threshold...decrementing said counter until zero, and sampling..."

Per claim 5:

-the software tool includes a profiler, compiler, assembler, interpreter, or virtual machine.

Arnold: Col. 2: 28, "compiler and interpreter run time processing environments Col. 2: 53, "virtual processor" (virtual machine)

Per claim 6:

-the interpreter includes a byte-code interpreter.

Arnold: Col. 2: 28, "compiler and interpreter run time processing environments Col. 2: 54,  
"JAVA threads assigned to it for execution" (byte code interpreter)

Per claim 7:

-the compiler includes one or more of an optimizing compiler and a byte code compiler.

Arnold: Col. 2: 28, "compiler and interpreter run time processing environments

Col. 2: 38-39, "applications that implement a 'compile-only' strategy by compiling all methods to native code before the execute" (byte code compiler)

Per claim 8:

-the predefined association is included within one or more of a compiler generated code, assembler generated code, an image, and an associated separate encoding.

Arnold: Col. 3: 10, "a yield point is defined as a special sequence of instructions Col. 3: 17, The compiler inserts yield points (compiler generated code) Col. 3: 35-36, placement of yield points in all method prologues and at the targets of all backwards intra-procedural branches...

Per claim 9:

Art Unit: 2191

-the instruction instance includes one or more of an instance of an instruction from a processor instruction set, an instance of an operation corresponding to a processor instruction set, an instance of a virtual machine instruction, or an instance of a byte code.

Arnold: Col. 3: 10, "a yield point is defined as a special sequence of instructions (instruction instance) Col. 3: 17, The compiler inserts yield points (compiler generated code) Col. 3: 35-36, placement of yield points in all method prologues and at the targets of all backwards intra-procedural branches... Col. 3: 40, take different sampling actions when a yield point is taken (corresponding instruction)

Per claim 10:

-the executable code includes one or more of object code, byte code, and machine code.

Arnold: Col. 3: 2, "compiled binary program"

Per claim 11:

-the source-level data object language construct representation includes one or more of a class, a data type, a data size, a data type definition, a data structure, linked object,, and a member of a data structure, static variables, automatic variables, memory segment.

Arnold: Col. 3: 36, method prologue (source level data object language construct)

Per claim 12:

Arnold failed to explicitly disclose:

-the corresponding language of the language construct includes a source-level language or an intermediate level language.

However Dean disclosed such details in the recorded state information, FIG. 3.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Arnold, using the teachings of Dean, because both references are directed towards profiling for optimization, using identified statistical analysis, related to cache misses / coherency, receiving feedback to improve program performance.

Per claim 13:

-the software tool aggregating runtime events based on the source-level data objects.

Arnold: Col. 5: 23-29, "...frequency of executed methods being recorded for characterizing program behavior...information such as the call context, frequency of executing basic blocks and program variable values may be recorded..." Col. 5: 33-34, "which method was executing...and incrementing a counter associated with that method" Col. 5: 59-60, collects a statistical sample of all method invocations Col. 5: 67, "tally the number of times it is invoked..."

Art Unit: 2191

Per claim 14:

-displays aggregated runtime events.

Arnold: Col. 6: 30-31, "the system will later process the statistical information collected as needed.

Per claim 15:

-aggregates profile data for the code based on the source-level data object language construct representation.

Arnold: Col. 5: 23-29, "...frequency of executed methods being recorded for characterizing program behavior...information such as the call context, frequency of executing basic blocks and program variable values may be recorded..." Col. 5: 33-34, "which method was executing...and incrementing a counter associated with that method" Col. 5: 59-60, collects a statistical sample of all method invocations Col. 5: 67, "tally the number of times it is invoked..."

Per claim 16:

A method of profiling code, the method comprising:

-for a sampled runtime event detected in execution of the code, identifying a corresponding operation instance of the code,

Art Unit: 2191

-and based on a predefined association between the identified operation instance and a language construct of a source-level representation corresponding to a source-level data object, attributing the detected event to the source-level data object.

See rejection of limitations addressed in claim 1 above.

Per claims 17-21:

See rejection of limitations addressed in claims 8-12 respectively above.

Per claim 22:

See rejection of limitations addressed in claim 3 above.

Per claim 23:

See rejection of limitations addressed in claim 4 above.

Per claim 24:

-embodied in a computer program product encoded on one or more machine-readable media.

Art Unit: 2191

Arnold: Col. 5: 10-11, architectures with a count register and associated machine instructions

Col. 6: 7-8, The trigger is implemented by reserving a single bit in the computer system's memory (machine readable media) Col. 6: 23, method count table

Per claim 25:

See rejection of limitations addressed in claim 15 above.

Per claim 26:

A method of profiling code, the method comprising:

-associating a language construct for a source-level data object with an instance of an instruction;

-based on the association between the source-level data object language construct and the instruction instance, attributing to the source-level data object language construct a sampled runtime event, which corresponds to the instruction instance, detected from execution of the code.

See rejection of limitations addressed in claim 1 above.

Per claim 27 & 28:

See rejection of limitations addressed in claims 10 & 11 above.

Per claim 29:

-the instruction instance includes a load type instruction.

Arnold: Col. 4: 17-40, counter based approach, pseudo code instructions. See sample code –  
load value into yield point counter

Per claim 30:

-the language construct includes one or more lexical tokens.

Arnold: Col. 4: 17-40, counter based approach, pseudo code instructions. See sample code – for  
examples of lexical tokens (if, else, (), =, ;, etc.)

Per claim 31:

-the lexical tokens include one or more of identifiers and literals.

Arnold: Col. 4: 17-40, counter based approach, pseudo code instructions. See sample code – for  
examples of lexical tokens / identifiers / literals (yieldPointCounter)

Per claim 32:

Art Unit: 2191

See rejection of claim 12 above.

Per claim 33:

See rejection of limitations addressed in claim 3 above.

Per claim 34:

Arnold failed to explicitly disclose:

-cache miss includes a data cache miss, an instruction cache miss, a unified cache miss, and an external cache miss.

However, Dean disclosed such useful trigger functions at col. 4: 33-48, col. 5: 7-21, col. 7: 35-43 (broadly – “choose a specific cache region to monitor”), col. 9: 29-36.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Arnold, using the teachings of Dean, because both references are directed towards profiling for optimization, using identified statistical analysis, related to cache misses / coherency, and receiving feedback to improve program performance.

Per claim 35:

Art Unit: 2191

Arnold failed to explicitly disclose:

-cache references includes one or more of a data cache reference, an instruction cache reference, a unified cache reference, and an external cache reference.

However, as an example Dean disclosed such limitations at col. 4: 39, "dirty data evictions (from the cache memory region)", col. 5: 7-21.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Arnold, using the teachings of Dean, because both references are directed towards profiling for optimization, using identified statistical analysis, related to cache misses / coherency, and receiving feedback to improve program performance.

Per claim 36:

See rejection of claim 4 above.

Per claim 37:

Arnold failed to explicitly disclose:

-backtracking from a second instruction instance to the instruction instance after detecting the sampled runtime event.

Art Unit: 2191

However Dean disclosed collecting state information and execution history (col. 13: 56-62) to identify the execution path taken to get to a particular instruction...knowledge of a second PC value of an instruction that was recently executed...

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Arnold, using the teachings of Dean, because both references are directed towards profiling for optimization, using identified statistical analysis, related to cache misses / coherency, and receiving feedback to improve program performance. To improve on the imprecise ambiguities (Dean - col. 13: 41-46) the above technique can aid in identifying the execution path.

Per claim 38:

See rejection of claim 24 above.

Per claim 39:

See rejection of claims 13 & 15 above.

Per claim 40:

Art Unit: 2191

A computer program product for profiling code, encoded in one or more computer readable media, the computer program product, when executed, performs operations comprising:

- identifying an instruction instance that corresponds to a runtime event;
- attributing the runtime event to a language construct representation of a source-level data object, the language construct representation having been associated with the instruction instance.

See rejection of limitations addressed in claim 1 above.

Per claim 41:

-the instruction instance includes one or more of a microinstruction and a macroinstruction.

Arnold: Col. 3: 2, "compiled binary program" Col. 4: 42-43, hardware architectures such as the PowerPc (instruction set specific to architecture maps to microinstructions)

Per claims 42 & 43:

See rejection of limitations addressed in claims 2 & 3 above.

Per claim 44:

See rejection of limitations addressed in claim 11 above.

Per claim 45:

See rejection of claim 15 above.

Per claim 46:

An apparatus comprising:

a computer readable encoding corresponding to an instruction sequence;  
means for correlating a sampled runtime event and a language construct that corresponds to a source-level data object that has an association with an instance of an instruction of the instruction sequence, the instruction instance corresponding to the sampled runtime event.

See rejection of limitations addressed in claim 1 above.

Per claims 47 & 48:

Regarding: computer readable encoding encodes a source-level representation corresponding to the instruction sequence, computer readable encoding encodes the instruction sequence.

See rejection of limitations addressed in claims 1 & 24 above.

Per claim 49:

Art Unit: 2191

An apparatus comprising:

a set of one or more processors;

a memory coupled with the set of processors;

a machine-readable media coupled with the set of processors, the machine-readable media having stored therein a set of data profiling instructions to cause the set of processors to, attribute a runtime event to a language construct that corresponds to a source-level data object that is associated with an instance of an instruction, which corresponds to the runtime event.

See rejection of limitations addressed in claims 1 & 24 above.

Per claim 50:

Arnold failed to explicitly disclose:

-at least one of the set of processors includes a data cache.

However, Dean disclosed (col. 4: 35-39) a particular cache...dirty data evictions... (modified data currently stored in a data cache will be evicted prior to subsequent use) Col. 6: 66-67, memory locations where the data are not in the cache (data cache miss).

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Arnold, using the teachings of Dean, because both references are directed

towards profiling for optimization, using identified statistical analysis, related to cache misses / coherency, and receiving feedback to improve program performance.

Per claim 51:

See rejection of claim 24 above.

Per claim 52:

See rejection of limitations addressed in claim 15 above.

***Conclusion***

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached at (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned: 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman  
12/06/2007

**MARY STEELMAN**  
**PRIMARY EXAMINER**

